Report Automation in Institutional Research

Richard Hetherington
Merima Babic
Courtney Raeford

temple.edu/ira
# Table of Contents

NEAIR 2015 Presentation – Report Automation in Institutional Research........ 2
Oracle Client Install Instructions ................................................................. 14
Visual Basic.NET Code ............................................................................. 16
Resources ................................................................................................. 20
Visual Basic.NET ...................................................................................... 20
Structured Query Language (SQL)............................................................ 20
Excel........................................................................................................... 20
WHY TEMPLE

OURS IS AN UNCOMMON DRIVE.
It’s what brings us together and sets us apart. It’s what leads us everywhere we want to go.
Report Automation in Institutional Research

Rich Hetherington
Associate Director

Merima Babic
Research Analyst

Courtney Raeford
Research Analyst

Data Analysis and Reporting
Overview

- Automation of cyclic reporting
  - VB.net application
  - Excel templates (VLOOKUP)
  - Structured Query Language (SQL)
- Process to our solution
- Demonstration of automated report
- Adaptability to IR offices
Temple University At-A-Glance

- Located in Philadelphia, Pennsylvania
- 18 Schools/Colleges
  - 5 Professional Schools
- 9 Campuses (including Rome & Japan)
- 450+ Academic Programs
- 38,000+ Students
- Approximately 3,500 Faculty
- www.temple.edu/ira
Why Report Automation?

- Volume of reports
- Consistency
- Resources

Goal:

- More time for analytical research
Design Constraints

Streamline and Automate Cyclic Reporting while having:

- Data consistency
- Process flexibility
- Self Sufficiency
- Transparency
- Scalability
Design Constraints

- Excel Output
- PDF Output
- Existing SQL code
- Ability to schedule
- Uncomplicated configuration
- Existing Excel Templates
- Minimal IT Unit support
Automation Process

Overview

Temple At-A-Glance

Why Automation?

Automation Process

Design Constraints

Demonstration

Conclusion

Questions

Windows Task Scheduler

VB.net Application (.EXE) → Database Connection → Send SQL Request → Set resulting table to variable

Open Excel Template → Go to Worksheet (“Data”) → Paste Data Result → Refresh Report Worksheet (VLOOKUPS)

Save Excel File → Create and Save PDF → Send Notification

Report Output

NEAIR 2015: Report Automation in IRA

Institutional Research & Assessment
Automation Process

TEMPE UNIVERSITY
Office of the Provost

Fall 2015 Enrollment Summary Report

(Summary includes all students registered for at least one on-campus course. Students registered at TU Online are not included.)

Institutional Research and Assessment

<table>
<thead>
<tr>
<th>Class Level</th>
<th>Full Time Undergraduate</th>
<th>Percent Undergraduate</th>
<th>Fall 2013</th>
<th>Fall 2014</th>
<th>Fall 2015</th>
<th>1 Year Change</th>
<th>2 Year Change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>37,611</td>
<td>37,713</td>
<td>37,634</td>
<td>0.26%</td>
<td>0.26%</td>
<td>0.26%</td>
</tr>
<tr>
<td></td>
<td>Freshman</td>
<td>7,354</td>
<td>7,416</td>
<td>7,418</td>
<td>0.08%</td>
<td>0.08%</td>
<td>0.08%</td>
</tr>
<tr>
<td></td>
<td>Sophomore</td>
<td>8,673</td>
<td>8,735</td>
<td>8,833</td>
<td>0.68%</td>
<td>0.68%</td>
<td>0.68%</td>
</tr>
<tr>
<td></td>
<td>Junior</td>
<td>10,218</td>
<td>10,279</td>
<td>10,418</td>
<td>0.59%</td>
<td>0.59%</td>
<td>0.59%</td>
</tr>
<tr>
<td></td>
<td>Master's Degree</td>
<td>1,994</td>
<td>2,046</td>
<td>2,194</td>
<td>2.60%</td>
<td>2.60%</td>
<td>2.60%</td>
</tr>
<tr>
<td></td>
<td>Graduate Non-Degree (Other)</td>
<td>2,935</td>
<td>3,007</td>
<td>3,155</td>
<td>2.38%</td>
<td>2.38%</td>
<td>2.38%</td>
</tr>
<tr>
<td></td>
<td>Undergraduate Non-Degree</td>
<td>2,375</td>
<td>2,404</td>
<td>2,496</td>
<td>1.20%</td>
<td>1.20%</td>
<td>1.20%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>37,611</td>
<td>37,713</td>
<td>37,634</td>
<td>0.26%</td>
<td>0.26%</td>
<td>0.26%</td>
</tr>
</tbody>
</table>

NEAIR 2015: Report Automation in IRA

Institutional Research & Assessment
Demonstration
Conclusion

- Scalability
- Scheduling
- Distribution
- Leverage Resources
  - In-office skills
  - Institutional Resources
  - Online Resources
- Efficiency
- Consistency
Questions?

A BREAKTHROUGH YEAR FOR TEMPLE UNIVERSITY

STATE-OF-THE-ART LIBRARY CONSTRUCTION TO START THIS SCHOOL YEAR

59% HONORS FRESHMEN: ALL-TIME RECORD

3.51 HIGHEST AVERAGE HIGH SCHOOL GPA

30,037 RECORD APPLICATIONS

53 NEW TENURED OR TENURE-TRACK FACULTY THIS YEAR

6 MEMBERS OF THE NATIONAL ACADEMIES

93% SIGN-UP FLY BY

Landscape master plan: More green space, more social space

17% PERCENT RISE IN 4 YEARS: FRESHMEN IN TOP ONE-TENTH OF THEIR CLASS

6 AMERICAN ACADEMY OF ARTS AND SCIENCES FELLOWS

CAMPUS IMPROVEMENTS

STUDENT SPARK

FACULTY EXCELLENCE

Enhanced FACILITIES, record-breaking STUDENTS and superb FACULTY are driving UNPRECEDEDED MOMENTUM.
Oracle Client Install Instructions

- In order to connect to the oracle environment via Visual Studio, the Oracle Data Access Components (ODAC) must be installed. The version used by the visual basic code provided, can be downloaded from: http://www.oracle.com/technetwork/topics/dotnet/utilsoft-086879.html

- Once the ODAC client has been installed, add the reference to your project in Visual Studio. Under Project, select Add Reference.
- Navigate to the .NET tab and find the Oracle.DataAccess component – double click to add to your project.

- Next, return to the Add Reference dialog box, navigate to the COM tab, find the Oracle Data Control component – double click to add to your project.
Visual Basic.NET Code

1. Imports System.Text
2. Imports System.IO
3. Imports System.Xml
5. Imports Oracle
6. Imports System.Net
8. Imports System.Management
9. 'Must add References (.NET, Microsoft.Office.Interop.Excel)
11. Imports Microsoft.Office.Interop
12. Module Module1
13.
14. Sub Main()
15. Dim uglist1 As New ArrayList
16. Dim uglist2 As New ArrayList
17. 'Reads ini to get path and filename for sql, excel template file, Output filename and output directory
18. Dim myini As String
19. Dim aPath As String
20. Dim aName As String
21. Dim line4 As String = ""
22. 'Determine the location of the ini file based on exe path
23. aName = System.Reflection.Assembly.GetExecutingAssembly. GetModules()(0).FullyQualifiedName
24. aPath = System.IO.Path.GetDirectoryName(aName)
25. myini = aPath & "\Param.ini"
26.
27. Try
28. 'Open the file using a stream reader.
29. Using sr As New StreamReader(myini)
30. line4 = File.ReadAllText(myini)
31. End Using
32. Catch er As Exception
33. End Try
34. 'Parse out the components of the ini file and set to variables
35. Dim myparams As String = line4
36. Dim mypparts As String() = myparams.Split(";")
37. Dim mysqlin As String = mypparts(0)
38. Dim myxls As String = mypparts(1)
39. Dim myoutputfolder As String = mypparts(2)
40. Dim myfilename As String = mypparts(3)
41. Dim line As String = ""
42. Try
43. 'Open the sql file using a stream reader.
Using sr As New StreamReader(mysqlin)
        line = sr.ReadToEnd()
    End Using
Catch er As Exception
    Console.WriteLine("The file could not be read:")
End Try
'*****************************************************
'**** OPEN ORACLE CONNECTION *********
'*****************************************************
Dim orapass As String = My.Resources.ORA_PASS
Dim oradb As String = "Data Source=(DESCRIPTION=
   + 
   + "(ADDRESS=(PROTOCOL=TCP)(HOST=prd-rac1.erp.temple.edu)(PORT=1521))" _
   + 
   + "(CONNECT_DATA=(SERVICE_NAME=PROD8));" _
   + 
   + "User Id=ADD_ORACLE_USERNAME;Password= " & orapass & ";"
'*****************************************************
Dim conno As New OracleConnection(oradb)
conno.Open()
Dim sql As String
Dim cmdo As New OracleCommand
Dim slbcount As Integer
'****** check for connection ***********
If (conno.State = 0) Then
    MsgBox("Oracle Connection Failure")
Else
    MsgBox("open")
End If
'This captures the SQL and passes it to Oracle
'sql = ""
sql = line
cmdo.CommandType = CommandType.Text
cmdo.CommandText = sql
    cmdo.Connection = conno
    Dim dro As OracleDataReader = cmdo.ExecuteReader()
'Passes sql into oracle connection - waits for returning data
While dro.Read()
    If dro.HasRows = False Then
        MsgBox("No Data Selected")
        GoTo SKIPifblank
    Try
    Catch ex As Exception
        'Console.WriteLine(ex.Message)
        Return
    End Try
End If
    END LOOP
END IF
    SKIPifblank:
    dro.Dispose()
96.  slbcount = uglist1.Count
97.  cmdo.Dispose()
98.  conno.Dispose()
99.  'Load sql output to excel template file
100. Dim xlApp As New Excel.Application
101. Dim xlWorkBook As Excel.Workbook
102. Dim xlWorkSheet As Excel.Worksheet
103. Dim xlSourceRange As Excel.Range
104. Dim xlDestRange1 As Excel.Range
105. Dim xlDestRange2 As Excel.Range
106. Dim myxlfile As String
107. myxlfile = myxls
108. xlWorkBook = xlApp.Workbooks.Open(myxlfile)
109. 'Display Excel
110. xlApp.Visible = True
111. 'Set the source worksheet
112. xlWorkSheet = xlWorkBook.Sheets("Data")
113. 'Set the source range
114. xlSourceRange = xlWorkSheet.Range("A2:B455")
115. 'Clear current data
116. xlSourceRange.Delete()
117. 'Set the destination range
118. xlDestRange1 = xlWorkSheet.Range("A2:A455")
120. xlDestRange2 = xlWorkSheet.Range("B2:B455")
122. xlWorkSheet = xlWorkBook.Sheets("REPORT")
123. xlWorkSheet.Activate()
124. Dim mydatestamp As String = DateTime.Now.ToString("yyyy-M-d")
125. Dim myexceltitle As String = myfilename & "." & mydatestamp & ".xlsx"
126. xlWorkSheet.SaveAs(myoutputfolder & myexceltitle)
127. xlWorkSheet.ExportAsFixedFormat(xlFixedFormatType.xlTypePDF, myoutputfolder & myexceltitle, Excel.XlFixedFormatQuality.xlQualityStandard, True, False, 1, 10, False)
128. xlWorkSheet.ExportAsFixedFormat(xlFixedFormatType.xlTypePDF, myoutputfolder & myfilename & "." & mydatestamp, Excel.XlFixedFormatQuality.xlQualityStandard, True, False, 1, 10, False)
129. xlWorkBook.Close()
130. xlApp.Quit()
Private Sub ReleaseObject(ByVal obj As Object)
    'This function is required in order to close all files and connections
    Try
        Dim intRel As Integer = 0
        Do
        Loop While intRel > 0
        'MsgBox("Final Released obj # " & intRel)
        Catch ex As Exception
            'MsgBox("Error releasing object" & ex.ToString)
            obj = Nothing
        Finally
            GC.Collect()
        End Try
    End Sub
End Module
Resources

Visual Basic.NET

For more information about connecting to an oracle database environment using Visual Studio:
http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/dotnet/GettingStartedVBVersion/Gettin gStartedNET_VBVersion.htm

For more information about How to automate Microsoft Excel from Visual Basic .NET:
https://support.microsoft.com/en-us/kb/301982

Visual Basic Quick Syntax Reference:

Microsoft's Visual Basic Programming Guide:
https://msdn.microsoft.com/en-us/library/y4wf33f0(v=vs.100).aspx

Structured Query Language (SQL)

Oracle SQL Online Language Guide:
https://docs.oracle.com/cloud/latest/db112/SQLRF/toc.htm

For Microsoft SQL Server environments, the following resources can assist with connection strings and syntax:
https://www.connectionstrings.com/sql-server/
https://support.microsoft.com/en-us/kb/308656

Excel

VLOOKUP Reference:
https://support.office.com/en-us/article/VLOOKUP-function-0bbc8083-26fe-4963-8ab8-93a18ad188a1